A Capsule based Approach for Polyphonic Sound Event Detection

Yaming Liu, Jian Tang, Yan Song, Lirong Dai
National Engineering Laboratory for Speech and Language Information Processing,
University of Science and Technology of China, Hefei, China
{lym66, enjtang}@mail.ustc.edu.cn, {songy, lrdai}@ustc.edu.cn

***Abstract*—Polyphonic sound event detection (polyphonic SED) is an interesting but challenging task due to the concurrence of multiple sound events. Recently, SED methods based on convolutional neural networks (CNN) and recurrent neural networks (RNN) have shown promising performance. Generally, CNN are designed for local feature extraction while RNN are used to model the temporal dependency among these local features. Despite their success, it is still insufficient for existing deep learning techniques to separate individual sound event from their mixture, largely due to the overlapping characteristic of features. Motivated by the success of Capsule Networks (CapsNet), we propose a more suitable capsule based approach for polyphonic SED. Specifically, several capsule layers are designed to effectively select representative frequency bands for each individual sound event. The temporal dependency of capsule's outputs is then modeled by a RNN. And a dynamic threshold method is proposed for making the final decision based on RNN outputs. Experiments on the TUT-SED Synthetic 2016 dataset show that the proposed approach obtains an F1-score of 68.8% and an error rate of 0.45, outperforming the previous state-of-the-art method of 66.4% and 0.48, respectively.**

I. Introduction

Sound event detection (SED), also known as acoustic event detection, aims at detecting the onset and offset times of sound events automatically and giving a label to each event. With the help of SED technology, computers can understand the environment around via sound and response to it. Recently, SED has received increasing interests due to its promising future with wide range of applications in our daily life, including acoustic surveillance [1], [2], bio-acoustical monitoring [3] and smart facilities in intelligent buildings [4].

According to whether SED allows multiple sound events to occur simultaneously, it can be categorized into monophonic and polyphonic ones, For monophonic SED, there exists a certain pattern for each individual sound event in spectrogram. For example, the rain event always fills the entire frequency bands, while the thunder event appears at low frequency bands. However, for polyphonic SED task, these patterns are very likely to overlap, which make it difficult to effectively separate individual pattern and make a correct detection for each event.

Traditional approaches for polyphonic SED include hidden Markov model (HMM) [5] and non-negative matrix factorization (NMF) [6]. In [6], NMF is used to separate the audio signal into 4 single tracks, where each track represents a combination of the original sources. This can be seen as a coarse separation of sound events. Recently, feedforward neural networks (FNN) and convolutional neural networks (CNN) have been successfully applied to audio event classification [7], [8] as well as polyphonic SED [9], [10]. Recurrent neural networks (RNN) [11] have achieved quite good performance by integrating information from the earlier time context. In [12], the CRNN which combines the strength of both CNN and RNN has obtained state-of-the-art polyphonic SED performance.

In this paper, we propose a capsule based approach for polyphonic SED, as shown in Fig. 1. This is motivated by Capsule Networks (CapsNet) [13], which have shown promising results on highly overlapped digital numbers classification. CapsNet are designed to predict the entire entity through partial information and to select suited predictions for the final classification. This characteristic may be useful for polyphonic SED task to separate each individual sound event from overlapped features of the mixture. In this work, firstly, a stack of convolutional layers are designed to extract local features from the input log mel band energies. Then the outputs of CNN are fed into two capsule layers, where local features from different frequency bands and channels are selected to predict multiple objects. A RNN is further applied to model the temporal dependency of capsule layers's outputs. To learn effective capsule representation, capsule layers and recurrent layers are jointly trained with two different loss functions concurrently. Compared with the existing deep learning based polyphonic SED methods, the proposed capsule based approach can effectively select representative frequency bands for each individual sound event, which is more suitable for separating sound events from their mixture. The performance is further improved by using a dynamic threshold according to validation metrics. Experiments on the TUT-SED Synthetic 2016 dataset show that the proposed approach obtains an improvement over the previous state-of-the-art method.

The main contributions of this study can be summarized as following:

- A capsule based framework is proposed for polyphonic SED to alleviate the overlap problem.
- A dynamic threshold strategy is used to make the final classification decision. This is a simple but effective decision method for polyphonic SED task.
- We experimentally demonstrate the validity of the proposed approach and analyze how events separate from their mixture through visualization.

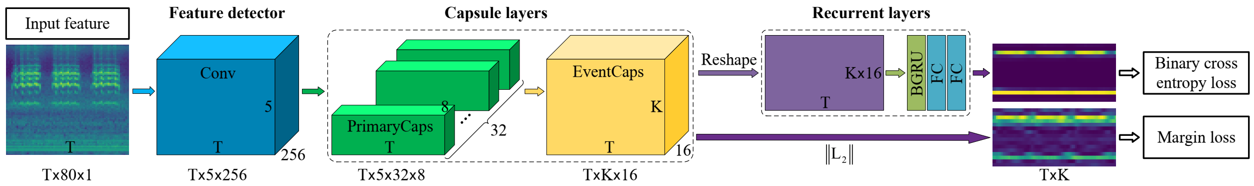

Fig. 1. Overall architecture of the proposed approach, which consists of three parts. 1) Feature detector: a group of convolutional layers with pooling only on frequency axis. 2) Capsule layers: the outputs of convolutional layers are fed into two capsule layers. 3) Recurrent layers: a bidirectional GRU and two FC layers are used to learn temporal context information and estimate event activity probabilities.

II. Method

A. Overview

The proposed framework illustrated in Fig. 1 includes three parts as follows. 1) Feature detector, which is composed of several convolutional layers with pooling only on frequency axis (time axis does not shrink), time-frequency representations of audio signal are fed into the detector. 2) Capsule layers, including a PrimaryCaps and an EventCaps, which are designed to select features from different frequency bands and channels. 3) Recurrent layers, which are used to learn temporal context information and estimate event activity probabilities. Hyperparameters used are presented in Table I.

B. Feature detector

We use 4 convolutional layers to detect local features in this work. Max-pooling is used to reduce frequency dimensionality, while time axis keeps the same to match the length of target. Log mel band energies $\mathcal{X} \in \mathbb{R}^{F\times T}$ is fed into the feature detector with zero-padding, where F is frequency bins of input features, T is the number of frames in a sample. The output of feature detector is a tensor $\mathcal{H} \in \mathbb{R}^{M\times F^{'}\times T}$, where M is the number of feature maps in the last convolutional layer, $F^{'}$ is the number of frequency bands after series of pooling operations.

C. Capsule layers

Capsules are vectors whose dimensions are associated with various properties of objects, such as location, size, orientation, etc. The length of each vector represents the activity probability of a specific object, and is limited to range from 0 to 1 by a nonlinear squashing function in (3). Two capsule layers are used in this work, a PrimaryCaps and an EventCaps. PrimaryCaps is a convolutional capsule layer with 32 channels. Each channel consists of 8D capsules. These capsules are also called low-level capsules which are fed into EventCaps later to obtain high-level ones. In EventCaps, firstly, prediction vectors of high-level capsules are calculated by multiplying outputs of low-level capsules by a weight matrix, as in (1). Then these prediction vectors are selected by routing-by-agreement process according to similarity between each high-level capsule's output and its prediction vectors, as in (2)(4)(5). The more similar a prediction vector and its corresponding high-level capsule's output are, the larger the connection weight between them is. This weight gain further increases the contribution of that prediction vector to its corresponding high-level capsule.

Let $\boldsymbol{u}_i$ denote the output of low-level capsule i, and $\boldsymbol{v}_j$ is the output of high-level capsule j, then $\boldsymbol{v}_j$ can be calculated as following

$$\hat{\boldsymbol{u}}_{j|i} = \boldsymbol{W}_{ij}\boldsymbol{u}_i \tag{1}$$

$$\boldsymbol{s}_j = \sum_i c_{ij}\hat{\boldsymbol{u}}_{j|i} \tag{2}$$

$$\boldsymbol{v}_j = \frac{\|\boldsymbol{s}_j\|^2}{1+\|\boldsymbol{s}_j\|^2}\frac{\boldsymbol{s}_j}{\|\boldsymbol{s}_j\|} \tag{3}$$

where $\hat{\boldsymbol{u}}_{j|i}$ is the prediction vector of high-level capsule j from low-level capsule i, $\boldsymbol{W}_{ij}$ is the corresponding weight matrix. The coupling coefficients c_{ij} are determined by dynamic routing process as follows

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \tag{4}$$

$$b_{ij} \leftarrow b_{ij} + \hat{\boldsymbol{u}}_{j|i}\cdot\boldsymbol{v}_j \tag{5}$$

where b_{ij} are log prior probabilities that low-level capsule i should be coupled with high-level capsule j. The b_{ij} are initialized to 0 and updated by the similarity between prediction vector $\hat{\boldsymbol{u}}_{j|i}$ and high-level capsule's output $\boldsymbol{v}_j$. This similarity can be measured by a scalar product operation.

Finally, margin loss of each output capsule k is calculated

$$L_k = T_k\max(0, m^+ - \|v_k\|)^2 + \lambda(1-T_k)\max(0, \|v_k\| - m^-)^2 \tag{6}$$

where T_k is 1 when class k actually exists, otherwise 0. Terms m^+, m^- and λ are hyperparameters, which are set to the same values as origin CapsNet [13]. The total loss is the sum of the losses of all output capsules.

In this work EventCaps calculates results on each frame simultaneously. The main process of capsule layers part is described as follows.

- PrimaryCaps, each channel contains $F^{'}\times T$ 8D capsules, i.e., $F^{'}\times 32$ primary capsules for each frame.
- Capsules of each frame are fed into EventCaps to compute K 16D capsules, where K is the number of event classes. Each 16D capsule represents one sound event. The output of EventCaps is a tensor $\mathcal{J} \in \mathbb{R}^{16\times K\times T}$.
- Calculating length of each capsule inside each frame. The output is a tensor $\mathcal{K} \in \mathbb{R}^{K\times T}$.

TABLE I
HYPERPARAMETERS USED IN THE PROPOSED APPROACH

	Feature detector				Capsule layers		Recurrent layers		
	Conv1	Conv2	Conv3	Conv4	PrimaryCaps	EventCaps	GRU	FC	FC
kernel	256@3 × 3	256@3 × 3	256@3 × 3	256@3 × 3	32@3 × 3	-	-	-	-
stride	1 × 1	1 × 1	1 × 1	1 × 1	1 × 1	-	-	-	-
pooling size	-	1 × 4	1 × 2	1 × 2	-	-	-	-	-
activation function	ReLU	ReLU	ReLU	ReLU	squashing	squashing	-	ReLU	sigmoid
num of hidden units	-	-	-	-	-	-	256	512	16
dim of capsule	-	-	-	-	8	16	-	-	-

D. Recurrent layers

A RNN is used to learn temporal context information, since the temporal dependency has proved to be important in sound event analysis task [11], [12], [14], [15], [16]. We reshape the output tensor of DigitCaps $\mathcal{J} \in \mathbb{R}^{16\times K\times T}$ to $\mathcal{M} \in \mathbb{R}^{(16\times K)\times T}$, i.e. combining K 16D capsules in each frame. These combined vectors are fed into a bidirectional gated recurrent unit (GRU). The bidirectional GRU outputs hidden state $\mathbf{h}_t$ at each frame t, followed by a feedforward layer with ReLU activation function. Finally, another feedforward layer with sigmoid activation function is used as the output layer. The output of recurrent layers is a tensor $\mathcal{F} \in \mathbb{R}^{K\times T}$, representing events activity probabilities of K sound events along T frames.

E. Dynamic Threshold

Event activity probabilities are binarized by a threshold to obtain binary representation, where 1 indicates an event exists and 0 indicates the opposite. The threshold can be very crucial to the performance of polyphonic SED system since we don't know how many events exist in one frame. In this work, an optimal threshold C_{opt} between [0.5, 0.9] is selected for each model using validation set. And the optimal threshold is used on the test set afterwards. We call this "dynamic threshold" since thresholds are different between models.

III. EXPERIMENTS

A. Datasets and Metrics

We evaluate the proposed approach on the dataset TUT-SED Synthetic 2016[1]. Audio in the dataset are artificially generated by randomly selecting and mixing isolated sound event samples from 16 sound event classes. The dataset consists of 100 mixed audio, each of which is 4-7 minutes. And total time of the dataset is around 10 hours. These audio are divided into three parts, 60% for training set, 20% for validation set and 20% for test set. All the audio are mono with 44.1 kHz sampling rate. Theoretically, the maximal number of sound events that occur simultaneously is 9, since that each audio is synthesized by 4-9 randomly selected classes.

Segment-based error rate (ER) and F1-score (F1) proposed in [17] are used as evaluation metrics in this work. Intermediate statistics are accumulated over the segments of the whole test set and then used to calculate ER and F1, which is called micro-averaging. To ensure comparability with the baseline system [12], we use two kinds of segments length, single frame (40ms) and one-second. Thus four evaluation metrics are used in this work, i.e., ER_{frame}, $F1_{frame}$, ER_{second} and $F1_{second}$, where ER_{frame} and $F1_{frame}$ as primary evaluation metrics.

B. Baseline

We compare our work with the previous state-of-the-art approach CRNN [12]. Work in [12] uses a CNN as feature extractor, log mel band energies is fed into the CNN. Time axis keeps the same during convolution, and feature maps are stacked along the frequency axis afterwards. These stacked feature maps are fed into a GRU later. Feedforward layer with sigmoid activation function is used as the output layer to obtain sound event activity probabilities of each frame.

C. Experiment Setting

We use the same log mel band energies as [12], except that the 80 mel bands are used instead. First a short-time Fourier transform (STFT) of 40ms and 50% overlap is applied to audio recordings to calculate spectrograms. Then an 80-bands mel filter bank is performed to compute mel band energies between 0 and 22050 Hz. After that, a log function is used to obtain log amplitude of mel band energies. Each mel band is normalized by subtracting its mean and dividing by its standard deviation calculated over the training set. At last, the normalized log mel band energies are split into samples by a sliding window with fixed length T (frames). Samples are overlapped during training and are nonoverlapping during validation and test, which is also the same as [12].

We run a hyperparameter grid search to select hyperparameters in this work, such as number of feature maps $\{64, 128, 256\}$ of feature detector; kernel size $\{(3,3), (5,5), (7,7)\}$ of convolutional layers including feature detector and PrimaryCaps layer; number of hidden units of BGRU and FC $\{128, 256, 512\}$ and sample frames T $\{64, 128, 256, 512, 1024\}$. At last, sample frame T is set to 128 to make a trade off between time consumption and validation metrics. Meanwhile, we also try different capsule dimension $\{(4,8), (8,16)\}$ and iteration times of routing $\{1, 2, 3, 4, 5\}$ (3 in this work). We finally use the hyperparameters shown in Table I, which obtain highest scores on the validation set.

All the networks are trained with Adam [18] optimizer with a fixed learning rate of 0.0001. Dynamic threshold strategy is

[1] http://www.cs.tut.fi/sgn/arg/taslp2017-crnn-sed/tut-sed-synthetic-2016

used in all our networks unless specifically mentioned. Batch normalization and dropout with dropout rate 0.25 are used after each convolutional layer. We use early stopping during the training process, holt the training if validation metrics are not improving for more than 10 epochs. The model with the best performance on validation set is chosen as the final model which is used to calculate results on test set. Each experiment is repeated 10 times with different random seeds to obtain more credible results.

The six comparison systems are as follows:

CNN is the first baseline from work [12], it consists three convolutional layers and no recurrent layer.

CRNN is the second baseline from work [12], including three convolutional layers and one GRU recurrent layer.

CapsNet is the CapsNet baseline which only includes feature detector and capsule layers. The model regards lengths of capsules of DigitCaps $\mathcal{K}$ as outputs. Margin loss is calculated between $\mathcal{K}$ and ground truth $\mathcal{Y} \in \mathbb{R}^{K \times T}$.

Capsule-RNN is our proposed approach as described in Section II. During the training process, two losses are calculated, i.e., binary cross-entropy loss between outputs of recurrent layers $\mathcal{F}$ and ground truth $\mathcal{Y}$, margin loss between capsules's lengths $\mathcal{K}$ and ground truth $\mathcal{Y}$. The final loss is a weighted sum of these two losses by a couple of weights [0.7, 0.3]. During the validation and test process, outputs of recurrent layers $\mathcal{F}$ are considered as event activity probabilities, and are binarized by dynamic threshold (DT).

Capsule-RNN without DT use a fixed threshold $C = 0.5$ (same as [12]) in Capsule-RNN to binarize event activity probabilities.

D. Results

In this section, we provide the mean and the standard deviation of F1 and ER in all experiments described above. As presented in Table II, CapsNet improves frame-based F1 and ER by relative 8% and 11% respectively compared with CNN baseline. Meanwhile, after adding temporal context information, Capsule-RNN also achieves a relative improvement of 4% on frame-based F1 and 6% on frame-based ER over CRNN baseline. Considering the number of parameters used for Capsule-RNN and CRNN are similar (CapsNet is even less than CNN), these improvements indicate an architectural advantage of capsule based methods compared with CNN based methods. On the other hand, compared with CapsNet, Capsule-RNN obtains a further 7% and 10% relative improvement on frame-based F1 and ER respectively, implying that the temporal context information is important to polyphonic SED task. This is consistent with previous work [12], [14]. Comparison between Capsule-RNN and Capsule-RNN without DT shows that a proper decision threshold is also helpful, especially for ER.

Fig. 2 shows an example from the test set for predicting events activities along 128 frames (2.56s), which also illustrates how the proposed approach is able to distinguish events from their mixture. We draw the coupling coefficients c_{ij} at the 80th frame (circled in red box) in Fig. 2-B. This distribution of coupling coefficients can be seen as a selection of frequency bands and channels when producing a high-level capsule from its prediction vectors. Two sound events are contained at the 80th frame, where *dog_barking* presents at middle and low frequency bands, and *glass_smash* covers all bands. Fig. 2 demonstrates that the proposed approach successfully detects these two events from their mixture by selecting channels on different frequency bands for *dog_barking* (the lowest frequency band) and for *glass_smash* (the second-highest frequency band), respectively.

TABLE II
ER AND F1 OF ONE FRAME SEGMENT BASED AND ONE SECOND SEGMENT BASED FOR BASELINES AND PROPOSED MODELS

Model	$F1_{frame}$	ER_{frame}	$F1_{second}$	ER_{second}
CNN [12]	59.8 ± 0.9	0.56 ± 0.01	59.9 ± 1.2	0.78 ± 0.08
CapsNet	64.6 ± 0.9	0.50 ± 0.01	65.0 ± 0.6	0.62 ± 0.01
CRNN [12]	66.4 ± 0.6	0.48 ± 0.01	68.7 ± 0.7	0.47 ± 0.01
Capsule-RNN	$\mathbf{68.8 \pm 0.7}$	$\mathbf{0.45 \pm 0.01}$	$\mathbf{69.2 \pm 0.5}$	$\mathbf{0.45 \pm 0.01}$
Capsule-RNN [a]	68.6 ± 0.8	0.47 ± 0.02	68.1 ± 0.8	0.51 ± 0.03

[a] Capsule-RNN without DT

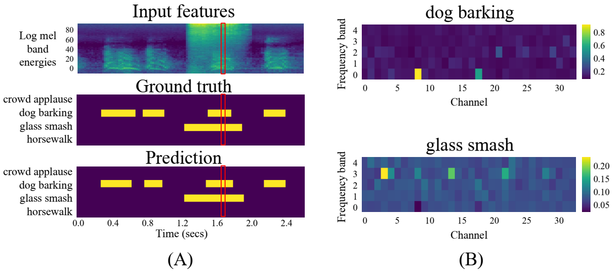

Fig. 2. (A) Input features, ground truth and prediction of an example from test set. (B) Coupling coefficients c_{ij} at the 80th frame. Each point in the image refers to the coupling coefficient between a high-level capsule (*dog barking* or *glass smash*) and its prediction vector from a low-level capsule. The vertical axis can be considered as frequency bands between 0-22050 Hz (0 for lowest frequency band, 4 for highest frequency band), while the horizontal axis represents the channels.

IV. Conclusions

In this paper, we proposed a capsule based approach for polyphonic sound event detection (SED) to alleviate the overlap problem. In our approach, several capsule layers were designed to effectively select representative frequency bands for each individual sound event. Also, the dynamic threshold strategy was proposed for selecting an optimal threshold for each model. Experiments showed that the proposed approach outperformed the previous state-of-the-art CRNN method and demonstrated the efficiency of this selection mechanism to sound events detection.

Acknowledgment

This work was supported by National Key R&D Program of China (Grant No. 2017YFB1002202) and the Key Science and Technology Project of Anhui Province (Grant No. 17030901005).

REFERENCES

[1] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*. IEEE, 2007, pp. 21–26.

[2] J. Saraswathy, M. Hariharan, S. Yaacob, and W. Khairunizam, "Automatic classification of infant cry: A review," in *Biomedical Engineering (ICoBE), 2012 International Conference on*. IEEE, 2012, pp. 543–548.

[3] D. Stowell and D. Clayton, "Acoustic event detection for multiple overlapping similar sources," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on*. IEEE, 2015, pp. 1–5.

[4] M. A. Sehili, B. Lecouteux, M. Vacher, F. Portet, D. Istrate *et al.*, "Sound environment analysis in smart home," in *International Joint Conference on Ambient Intelligence*. Springer, 2012, pp. 208–223.

[5] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, p. 1, 2013.

[6] T. Heittola, A. Mesaros, T. Virtanen, and A. Eronen, "Sound event detection in multisource environments using source separation," in *Machine Listening in Multisource Environments*, 2011.

[7] I. McLoughlin, H. Zhang, Z. Xie *et al.*, "Robust sound event classification using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 540–552, 2015.

[8] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 559–563.

[9] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–7.

[10] E. Cakir, E. C. Ozan, and T. Virtanen, "Filterbank learning for deep neural network based polyphonic sound event detection," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 3399–3406.

[11] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 6440–6444.

[12] G. Parascandolo, T. Heittola, H. Huttunen, T. Virtanen *et al.*, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[13] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, 2017, pp. 3859–3869.

[14] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. D. Plumbley, "Convolutional gated recurrent neural network incorporating spatial features for audio tagging," in *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 2017, pp. 3461–3466.

[15] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," *arXiv preprint arXiv:1710.00343*, 2017.

[16] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. Plumbley, "Attention and localization based on a deep convolutional recurrent model forweakly supervised audio tagging," *Proceedings of Interspeech 2017*, pp. 3083–3087, 2017.

[17] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[18] D. Kinga and J. B. Adam, "A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.